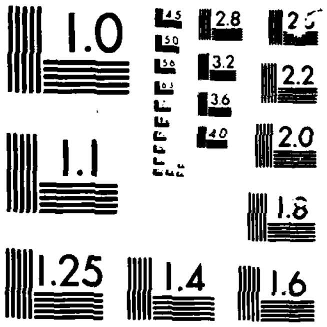
AD-A167 577 ADA (TRADEMARK) COMPILER VALIDATION SUMMARY REPORT: SOFTECH INC ADA86 COM..(U) FEDERAL SOFTWARE MANAGEMENT SUPPORT CENTER FALLS CHURCH VA 14 NOV 85 1/1

UNCLASSIFIED F/G 9/2 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

(2)

Ada* COMPILER VALIDATION SUMMARY REPORT:

SofTech, Inc.
Ada86 Compiler
Version 1.21
VAX-11/780, VAX-11/785;
Using VAX/VMS Version 4.1;
and the INTEL 8086 and the INTEL 80186

November 14, 1985

Prepared by:

Federal Software Management Support Center
Office of Software Development
and Information Technology
Two Skyline Place, Suite 1100
5203 Leesburg Pike
Falls Church, VA 22041-3467

Prepared for:

SofTech, Inc.
460 Totten Pond Road
Waltham, MA 02254

Ada Joint Program Office
1211 South Fern St.
Arlington, Va 22202

Version 0.3

* Ada is a registered trademark of the U.S. Government, (Ada Joint Program Office).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER / 2. GOVT ACCESSION NO. AD-A167577	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Ada Compiler Validation Summary Report: Softech, Inc. Ada86 Compiler, Version 1.21, VAX-11/780, VAX-11/785; Using VAX/VMS Version 4.1 and the INTEL 8086 and the INTEL 80186	5. TYPE OF REPORT & PERIOD COVERED 14 Nov. 1985 to 14 Nov. 1986
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Federal Software Management Support Center	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Federal Software Management Support Center Office of Software Development and Information Technology Two Skyline Palce, Suite 1100, 5203 Leesburg Pike, Falls Church, VA 22041-3467	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Ada Joint Program Office 1211 S. Fern St., Rm. C-107 Arlington, VA 22202	12. REPORT DATE 14 November 1985
	13. NUMBER OF PAGES 52
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Federal Software Management Support Center	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

UNCLASSIFIED

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Ada Programming language, Ada Compiler Validation Summary Report, Ada Compiler Validation Capability, ACVC, Validation Testing, Ada Validation Office, AVO, Ada Validation Facility, AVF, ANSI/MIL-STD-1818A, Ada Joint Program Office, AJPO.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See Attached.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

The purpose of this Validation Summary Report is to present the results and conclusions of performing standardized tests on the SofTech Ada86 compiler, Version 1.21. On-site testing was performed by the Federal Software Management Support Center (FSMSC)--an Ada Validation Facility-- in accordance with current Ada Validation Office policies and procedures, on 1 - 8 November, 1985 at SofTech, Inc. in Waltham, MA.

The suite of tests known as the Ada Compiler Validation Capability (ACVC), Version 1.6, was used. The ACVC suite of tests is used to validate conformance of the compiler to ANSI/MIL-STD-1815A (Ada). This standard is described in the ANSI Ada Reference Manual, January 1983. Not all tests in the ACVC test suite are applicable to this specific implementation. Also, known test errors in Version 1.6 are present in some tests; these tests were withdrawn. The purpose of the testing is to ensure that the compiler properly implements legal language constructs and that it identifies, rejects from processing, and labels illegal constructs.

The SofTech Compiler Ada86, Version 1.21, using VAX/VMS 4.1 was tested with version 1.6 of the ACVC validation tests. Version 1.6 of the test suite contains 2162 tests of which 63 were withdrawn and 219 were inapplicable to this implementation. All of the 1880 remaining tests were passed.

The host/target relationships are described in section 1.2 of this document.

[illegible]

Richard G. Harrison
Director
Federal Software Management Support Center

[illegible]

Dr. John F. Kramer, Jr.
Institute for Defense Analyses

Virginia L. Castor

Virginia L. Castor
Director
Ada Joint Program Office

TABLE OF CONTENTS

1. Introduction

1.1 Purpose of the Validation Summary Report

This report describes the results of the validation testing for the compiler designated as Ada86, Version 1.21 for the following configurations:

Host Machines:	VAX-11/780 and VAX-11/785.
Operating System:	VAX/VMS 4.1
Host Disk Systems:	RA81, RP06, RH81
Target Machines:	INTEL 8086, INTEL 80186
Operating System:	None
Language Version:	ANSI/MIL-STD-1815A Ada
Translator Name:	Ada86 1.21
Validation Test Version:	1.6

Testing of this compiler was conducted by the Federal Software Management Support Center under the supervision of the Ada Validation Office (AVO), at the direction of the Ada Joint Program Office. Testing was conducted from 1 - 8 November, 1985 at SofTech, Waltham, Mass. All testing was performed in accordance with AVO policies and procedures.

The purpose of this report is to document the results of the testing performed on the compiler, and in particular, to:

. identify any language constructs supported by the compilers that do not conform to the Ada standard;

. identify any unsupported language constructs required by the Ada standard; and

. describe implementation dependent behavior allowed by the standard.

1.2 Host To Target Relationship Table

The SofTech Ada86 Compiler was tested on those systems listed in the table below.(*) The following table represents the tested target/host relationships:

Host	Target	Series of Testing	Site of Execution
VAX-11/785	INTEL 8086	Full ACVC	Waltham, MA
VAX-11/785	INTEL 80186	Subset	Waltham, MA
VAX-11/780	INTEL 8086	Subset	Waltham, MA
VAX-11/780	INTEL 80186	Subset	Waltham, MA

(*) The configuration for testing under the VAX-11/785 was three Digital VAX-11/785 systems configured in a VAX DECNET architecture. The term VAX-11/785 will be used to reference the testing performed on the three systems configured under VAX DECNET.

The Intel 8086 chip was housed on the 86/30 board consisting of:

one 8087 numeric processor chip for floating point operations
one 8253 timer control chip
one 8259 interrupt controller chip

The Intel 80186 chip was housed on the 186/03A board consisting of:

one 80186 chip with the timer and interrupt controllers integrated within the chip itself.
one 8087 numeric processor chip for floating point operations

There is no provision in the compiler for floating point operations without the 8087 numeric processor chip.

1.3 Use of the Validation Summary Report

The Ada Validation Office may make full and free public disclosure of this report in accordance with the "Freedom of Information Act" (5 U.S.C. #552). The results of the validation apply only to the computers, operating systems, and compiler version identified in this report.

The Ada Compiler Validation Capability is used to determine insofar as is practical, the degree to which the subject compiler conforms to the Ada standard. Thus, this report is necessarily discretionary and judgemental. The United States Government does not represent or warrant that the statements, or any one of them, set forth in this report are accurate or complete, nor that the subject compiler has no other nonconformances to the Ada standard. This report is not meant to be used for the purpose of publicizing the findings summarized herein.

Any questions regarding this report or the validation tests should be sent to the Ada Validation Office at:

Ada Joint Program Office
1211 South Fern Street
Arlington, VA 22202

1.4 References

Reference Manual for the Ada Programming Language, ANSI/MIL-STD-1815A, February 1983.

Ada Validation Organization: Policies and Procedures, Mitre Corporation, June 1982, PB 83-110601.

Ada Compiler Validation Implementers' Guide, SofTech, Inc., October 1980.

The Ada Compiler Validation Capability, Computer, Vol. 14, No. 6, June 1981.

Using the ACVC Tests, SofTech, Inc. November 1981.

Ada Compiler Validation Plans and Procedures, SofTech, Inc. November 1981.

1.5 Definitions of Terms

Class A tests are passed if no errors are detected at compile time. Although these tests are constructed to be executable, no checks can be performed at run-time to see if the test objective has been met; this distinguished Class A from Class C tests. For example, a Class A test might check that keywords of other languages (other than those already reserved in Ada) are not treated as reserved words by an Ada implementation.

Class B tests are illegal programs. They are passed if all the errors they contain are detected at compile-time (or link-time) and no legal statements are considered illegal by the compiler.

Class L tests consist of illegal programs whose errors cannot be detected until link time. They are passed if errors are detected prior to beginning execution of the main program.

Class C tests consist of executable self-checking programs. They are passed if they complete execution and do not report failure.

Class D tests are capacity tests. Since there are no firm criteria for the number of identifiers permitted in a compilation, number of units in a library, etc., a compiler may refuse to compile a class D test. However, if such a test is successfully compiled, it should execute without reporting a failure.

Class E tests provide information about an implementation's interpretation of the Standard. Each test has its own pass/fail criterion.

ACVC:	Acronym for the Ada Compiler Validation Capability.
AVO:	The Ada Validation Office. In the context of this report the AVO is responsible for directing compiler validation.
CHECK or CHECKTEST:	An automated tool defined by the Federal Software Management Support Center (FSMSC) and developed by the AVF that produces summary test results by reading compiler output in a spool file.
CUSTOMER:	The agency requesting the validation SofTech, Inc. Waltham, MA.

FSMSC:	Federal Software Management Support Center. In the context of this report the FSMSC conducts Ada validations under contract to the AVO as a satellite facility.
HOST:	The computer on which the compiler executes.
IG:	ACVC Implementors' Guide.
RM:	The Ada Language Reference Manual.
STANDARD:	The standard for the Ada language, ANSI/MIL-STD-1815A (1983).
SUBSET TESTS:	A grouping of ACVC tests selected by the FSMSC. Each chapter in the ACVC is represented in the subset by between 4 to 7 tests. The subset is used for statistical sampling of the various host and target hardware configurations.
TARGET:	The computers for which the compiler generates object code.
VALIDATION:	The process of testing a compilation system to certify that it conforms to the standard.
VALIDATION TESTS:	The set of test programs used to detect non-conformances in compilation systems. In this report, the term will be used (unqualified) to mean the ACVC tests.

2. TEST ANALYSIS

The following table shows that the SofTech, Inc. AdaVAX compiler passed all applicable tests.

	A	B	C	D	E	L	Total
In suite	61	800	1273	17	8	3	2162
Inapplicable	0	9	209	0	1	0	219
Withdrawn	0	17	46	0	0	0	63
Passed	61	774	1018	17	7	3	1880
Failed	0	0	0	0	0	0	0

2.1 Class A Testing

Class A tests check that legal Ada programs can be successfully compiled. These tests are executed but contain no executable self-checking capabilities.

2.1.1 Class A Test Procedures

Each class A test was separately compiled and executed. However, the only purpose of execution is to produce a message indicating that the test passed.

2.1.2 Class A Test Results

Successful compilation and execution without any error messages indicates the tests passed. All applicable tests passed.

2.2 Class B Testing

Class B tests check the ability to recognize illegal language usage. All applicable class B tests passes.

2.2.1 Class B Test Procedures

Each Class B test was separately compiled. The resulting test compilation listings are manually examined to see whether every illegal construct in the test is detected. If some errors are not detected, a version of the program test is created that contains only undetected illegal constructs.

This revised version is recompiled and the results analyzed. If some errors are still not detected, the revision process is repeated until a revised test contains only a single previously undetected illegal construct.

A class B test is considered to fail only if a version of the test containing a single illegal construct is accepted by the compiler (i.e., an illegal construct is not detected) or a version containing no errors is rejected (i.e., a legal construct is rejected).

2.2.2 Class B Test Results

There were 800 class B tests presented to the compiler. Of these tests 9 were found to be inapplicable to this implementation (see Section 4.2.7); 17 tests were found to be incorrect (i.e., a conforming compiler would have failed each of these tests). All remaining class B tests passed.

Because all errors were not detected when compiling the original tests, the following tests were modified by removing the detected errors; the modified tests were then submitted again to see if the remaining errors would be detected.

```
B33004A.ADA  B37301B.ADA  B44001A.ADA  B45205A.ADA
B55A01A.ADA  B67001A.ADA  B67001B.ADA  B67001C.ADA
B67001D.ADA  BC10AEB.ADA  BC1202B.ADA  BC1202D.ADA
```

All illegal constructs were detected except in some tests that were withdrawn because of errors in the tests (see Section 4.2.6).

2.3 Class C Testing

Class C tests check that legal Ada programs are correctly compiled and executed by an implementation.

2.3.1 Class C Test Procedures

Each Class C test is separately compiled and executed. The tests are self-checking and produce PASS/FAIL messages. Any 'failed' tests are individually checked to see if they are correct and if they are applicable to the implementation. Any tests that are inapplicable or that do not conform to the Ada Standard are withdrawn.

2.3.2 Class C Test Results

All class C tests which were applicable passed.

2.4 Class D Testing

Class D tests are executable tests used to check an implementation's compilation and execution capacities.

2.4.1 Class D Test Procedures

Each class D test is separately compiled and executed. The tests are self-checking and produce PASS/FAIL messages.

2.4.2 Class D Test Results

All applicable class D tests passed.

2.5 Class E Testing

Class E tests provide information about an implementation's interpretation of the Standard where the Standard permits implementations to differ. Each test has its own pass/fail criterion.

2.5.1 Class E Testing Procedures

Each class E test is separately compiled and executed. The tests are self-checking and produce pass/fail messages.

2.5.2 Class E Test Results

Of the class E tests, 1 was found to be inapplicable for this validation. All of the remaining tests passed.

2.6 Class L Testing

Class L tests check that incomplete or illegal Ada programs involving multiple separately compiled source files are detected at link time and are not allowed to execute.

2.6.1 Class L Test Procedures

Each Class L test is separately compiled and execution is attempted. The tests produce FAIL messages if executed. Any "failed" tests are checked to see if they are correct and applicable to the implementation. Tests that are inapplicable or that do not conform to the Ada standard are withdrawn.

2.6.2 Class L Test Results

Of the class L tests, none were found to be inapplicable to this implementation, and none were withdrawn due to errors in the tests. All L tests passed.

2.7 Subset Testing

A subset of the executable ACVC tests was identified by the FSMSC and used during this validation. This subset of tests was used to test multiple configuration combinations during the validation, specifically, the VAX-11/780 as a host targeted to Intel 8086 and the Intel 80186 and the VAX-11/785 as a host targeting the INTEL the 80186.

The subset comprised the following tests:

Chapter 2	Chapter 3	Chapter 4	Chapter 5
C23001A	C34001A	C41101D	C51002A
C24102A	C34001H	C42005A	C52001A
C26008A	A32203D	C43214A	C53005A
A29002A	C35904A	C45101A	C54A03A
	C36204A	C48004A	D55A03A
	C34002B		

Chapter 6	Chapter 7	Chapter 8	Chapter 9
C61003B	A71002A	A83A02A	C92002A
A62006D	C72001B	C84002A	C93001A
C63004A	C74302A	C85007E	C94006A
C65003A	C74209A	C86003A	A97106A
C66002A	C74409B	C87B48A	C97202A

Chapter 10	Chapter 11	Chapter 12	Chapter 14
CA1003A	CB1001A	CC1004A	AE2101A
CA2004A0M	CB2004A	CC3004A	CE2102A
CA2004A1	CB3003A	CC3408A	CE2201A
CA2004A2	CB4001A	CC3504C	CE2401E
CA2004A3			CE3102A
CA2004A4			CE3901A

*** CZ ***

CZ1101A
CZ1102A
CZ1103A
CZ1201A
CZ1201B
CZ1201C
CZ1201D

3. COMPILER ANOMALIES AND NONCONFORMANCE

There were no nonconformances to the Ada standard detected in this validation. The compiler passed all applicable correct tests.

4. ADDITIONAL INFORMATION

This section describes in more detail how the validation was concluded.

4.1 Compiler Parameters

Certain tests do not apply to all Ada compilers, e.g., compilers are not required to support several predefined floating point types, and so tests must be selected based on the predefined types an implementation actually supports. In addition, some tests are parameterized according to the maximum length allowed by an implementation for an identifier (or other lexical element; this is also the maximum line length), the maximum floating point precision supported, etc. The implementation dependent parameters used in performing this validation were:

. maximum lexical element length: 120 characters.

. maximum digits value for floating point types: 15

. SYSTEM.MIN_INT: -2147483648

. SYSTEM.MAX_INT: 2147483647

. predefined numeric types: INTEGER, FLOAT, LONG_INTEGER and LONG_FLOAT.

INTEGER_FIRST: -32_768

INTEGER_LAST: 32_767

. LONG_INTEGER'FIRST: -2147483648

. LONG_INTEGER'LAST: 2147483647

. Source character set: ASCII

. Extended ASCII chars: abcdefghijklmnopqrstuvwxyz !$%?@[\]^'{}~"

. non-ascii char type: (NON_NULL)

. TEXT_IO.COUNT'LAST: 32_767

. TEXT_IO.FIELD'LAST: 32_767

. illegal external file name1: "bad-character*^"

. illegal external file name2: "much_too_long_name_for_a_file"

. SYSTEM.PRIORITY'FIRST: 1

. SYSTEM.PRIORITY'LAST: 15

4.2 Testing Information

Complete ACVC test runs were compiled/executed at SofTech, Inc., Waltham, MA.

4.2.1 Pre-Test Procedures

Prior to testing, appropriate values for the compiler-dependent parameters were determined. These values were used to adapt tests that depend on the values. A magnetic tape containing the adapted tests was prepared and brought to the testing site. Spilt B tests were not prepared on this tape and had to be split on-site.

For pre-validation, SofTech compiled the entire ACVC on the VAX-11/785 in Waltham, MA. and targeted the results to execute on the INTEL 8086.

4.2.2 Control Files

SofTech, Inc. provided command procedures that compiled and executed tests automatically.

4.2.3 On-site Data Collection

All test hardware was located in Waltham, MA. Two Intel 8086's were used to speed the testing.

All applicable tests in the ACVC were compiled on the VAX-11/785, downloaded to one of the Intel 8086's, and executed. The results from each test execution were loaded back up into the VAX-11/785 after execution, for storage. This sequence was then repeated for each applicable executable test in the ACVC.

Upon completion of the entire ACVC run above, the collective test results from this full ACVC run were compared to the complete ACVC run performed for pre-validation. The results were analyzed and found to be identical and correct.

Three identical but individual subsets of the ACVC tests (see 2.7) were compiled on both the VAX-11/780 and the VAX-11/785. The two subsets of load modules compiled on the VAX-11/780 were individually downloaded to one of the 8086's and the 80186 and executed. The subset load modules compiled on the VAX-11/785 were individually downloaded to the 80186 and executed. The compilation and execution results were compared and found to be identical and correct.

Linked relocatable object modules were relocated with a standard Intel relocation utility (LOC86) and downloaded to an Intel development system network using the Intel NDS-II/VAX Link. The loading and execution of the relocated programs was done under the control of an Intel Series III and IV development system equipped with an in-circuit emulator, to which the 86/30 and the 186/03A boards were attached.

4.2.4 Test Analysis Procedures

On completion of testing the base system, all results were analyzed for failed Class A, C, D, E, and L tests, and all class B compilation results were individually analyzed. Analysis procedures are described for each test class in chapter 2.

4.2.5 Timing Information

The elapsed times required for compiling the non-executable tests and compiling, linking, exporting and running the executable tests.

VAX 11/785 264:00(*) Full compile, download, and execution of the validation suite compiled on the VAX/VMS - batch streams in a VAX DECNET environment targeted to 2 Intel 8086 targets.

VAX 11/785	26:46	Subset of validation suite under VAX/VMS - targeted to an Intel 80186
VAX 11/780	24:18	Subset of validation suite under VAX/VMS targeted to an Intel 8086
VAX 11/780	20:40	Subset of validation suite under VAX/VMS targeted to an Intel 80186

(*) Elapsed time is sum of elapsed times on each of five pieces of hardware used: 3 VAX 11/785 and 2 Intel 8086's.

4.2.6 Description of Errors in Withdrawn Tests

The following tests in version 1.6 of the ACVC did not conform to the ANSI Ada standard and were withdrawn for the reasons given below:

. B66001A-B: Test checks (in section G) that a parameterless function that is equivalent to an enumeration literal in the same declarative region is a redeclaration and, as such, is forbidden. According to RM 8.3(17), the explicit declaration of such a function is allowed if an enumeration literal is considered to be an implicitly declared predefined operation. The RM is not clear on this point. This issue has been referred to the Language Maintenance Committee for resolution. Since the issue cannot be resolved at this time, the test is withdrawn from Version 1.6. (Please note that this test may be considered correct and may appear in the future Versions of the ACVC, including Version 1.6.)

. BC1013A-B: The declaration of equality in lines 86-87 is illegal because the parameter type T declared in line 11 is not a limited type (LRM 6.7-4).

. C45521A, C45521B, ... C45521Y (25 tests) : Cases C and I define the model interval for the result too narrowly.

. C48005C-B: Lines 38 and 63 of this test should check that the value of the designated object is null.

. C64103C-B: This test should raise CONSTRAINT_ERROR during the conversion at line 179.

. C64103D-B: This test involves a CONSTRAINT_ERROR vs. NUMERIC__ERROR issue that is to be resolved by the Language Maintenance Committee.

. C64105E-AB: For case E, ensure that non-null dimensions of formal and actual parameters belong to both index subtypes (see AI-00313).

. C64105F-AB: For case E, ensure that non-null dimensions of formal and actual parameters belong to both index subtypes (See AI-00313).

. B67001A-B: Line 414 is missing the "BEGIN NULL; END;" needed to complete the block beginning at line 389 (case H).

- B67004A-B: The default name for a formal generic equality function should not be allowed to be "/=" unless an expanded name is used.

- C93005A, C93005B, and C93005C: These tests contain a declaration of an interger variable whose initialization is solely for the purpose of raising an exception. Some compilers will not raise this exception due to their optimization.

- C93007B-B: This test should check for PROGRAM_ERROR rather than TASKING_ERROR (See AI-000149).

- CA1011A*-B: The test objective should be reversed to be consistent with AI-00199.

- CA1108A-B: A pragma ELABORATE is needed for OTHER_PKG at line 25.

- CA1108B-B: A pragma ELABORATE is needed for FIRST_PKG at line 39 and for LATER_PKG at line 49.

- CA2009B-B, CA2009E-B: The repetition of the main procedure after the subunit body makes the subunit body obsolete; therefore, an attempt to execute the main procedure will fail.

- CA2009F*-B: The file CA2009F2-B is missing from the test suite.

- BC3204A-B, BC3204B-B, BC3204C*-B, BC3204D-B, BC3205A-B, BC3205B-B, BC3205C-B, BC3205D*-B, BC3405B-B: Instantiations with types that have default discriminants are now legal (AI-00037).

- CE3603A-B: The last case is inconsistent with AI-00050. If the string argument is null, no attempt to read is made and END_ERROR is not raised.

- CE3604A-B: Cases 5, 8, 9, and 11 are inconsistent with AI-00050. SKIP__LINE is called only if the end of the output string has not been met.

- CE3704M-B: A superfluous SKIP_LINE causes the input and output operations to be out of synchronization.

- B38105B-AB: This test requires a specific interpretation of the Language Reference Manual (LRM) regarding whether an incomplete type can have discriminate contraints before the full type declaration; this interpretation is not fully supported by the LRM or Language Maintenance Committee.

- C48006B-B: This test requires a specific interpretation of the Language Reference Manual (LRM) regarding whether an incomplete type can have discriminate constraints before the full type declaration; this interpretation is not full fully supported by the LRM or Language Maintenance Committee.

- B74103F-B: This test hinges on whether or not a generic formal type declaration declares a type. This matter will be debated by the Language Maintenance Committee in November.

- B74207A-B: This test requires a specific interpretation of the Language Reference Manual (LRM) regarding whether an incomplete type can have discriminate constraints before the full type declaration; this interpretation is not fully supported by the LRM or Language Maintenance Committee.

- CA1003B-AB: A compilation that contains an illegal compilation unit may now be rejected as a whole (see AI-00255/05).

- BC3503A-B: This test requires a specific interpretation of the Language Reference Manual (LRM) regarding whether an incomplete type can have discriminate constraints before the full type declaration; this interpretation is not fully supported be the LRM or Language Maintenance Committee.

- CE2107E-B: This test has a variable, TEMP_HAS_TRUE, that needs to be given an initial value of TRUE.

4.2.7 Description of Inapplicable Tests

B52004E, B55B09D, B86001CR, C34001D, and C55B07B were inapplicable because the implementation does not support SHORT_INTEGER.

B86001CP, C34001F, and C35702A were inapplicable because the implementation does not support SHORT_FLOAT.

B86001DT was inapplicable because the implementation does not support LONG_LONG_INTEGER.

BA2001E is inapplicable because Ada RM 10 2/5.4 states that the "simple names of all subunits that have the same ancestor library unit must be distinct identifiers". The test expects that the above condition be checked at the point of the declaration of the stub. The implementation detects a duplicate subunit name under a single ancestor library unit when the subunit itself is being compiled. No program library will contain duplicate subunits since the second of the subunits will be rejected.

BC3009A, BC3009B, BC3009C are inapplicable because of detection of generic instantiation only on an instantiation of a "real" entity - an instantiation outside of a generic entity. Since no subprogram or package is instantiated outside of a generic unit in these tests, the circularity is not detected. In essence, since generics are treated as templates, only a "real" instantiation actually brings a copy into being: circularity within a template is tolerated through no instantiation of this template will be legal.

The following tests were inapplicable because they exceed the accuracy of the floating-point definition for the target implementation:

C35705L through C35705Y (14)
C35706L through C35706Y (14)
C35707L through C35707Y (14)
C35708L through C35708Y (14)
C35802L through C35802Y (14)
C45241L through C45241Y (14)
C45321L through C45321Y (14)
C45421L through C45421Y (14)
C45424L through C45424Y (14)
C45621L through C45621Z (15). (=141 tests, total)

C24113H through C24113Y were in applicable because the implementation does not support the line length specified in the tests. (18 tests)

C86001F was inapplicable because "SYSTEM" is recompiled, requiring all other library packages to be compiled.

C96005B - This test checks to find a difference between DURATION'BASE'FIRST and DURATION'FIRST. If no difference exists (as is the case in the implementation) the test is inapplicable.

CE2107B, CE2110B, CE2111D, CE3704F, CE3704N, CE3905L - implementation does not allow you to CREATE a file.

CE2107A - Test incorrectly reports failure when STATUS_ERROR results from attempts to DELETE unopened files.

CE2110A - Test incorrectly reports failure when USE_ERROR results from attempts to CREATE files.

CE2111A, CE2111B, CE2111C - Test terminates when unhandled USE ERROR results from attempt to CREATE a file.

CE2201B - Test incorrectly reports failure when its explicitly raised INCOMPLETES are handled as OTHERS.

CE2201C, CE2202A - Test incorrectly reports failure when USE ERROR results from attempts to CREATE files.

CE2401A, CE2401B, CE2401C, CE2401D, CE2401E, CE2402A, CE2409A - Test is illegal because POSITIVE_COUNT'LAST is 1.

CE2401F - Test is illegal because POSITIVE_COUNT'LAST is 1. Also, test incorrectly reports failure when USE_ERROR results from attempts to CREATE files, and also, test incorrectly reports failure STATUS__ERROR results from attempts to RESET unopened files.

CE2404A, CE2405B, CE2406A, CE2408A, CE2410A - Test terminates when unhandled USE_ERROR results from attempt to CREATE a file.

CE3102B - Test reports failure when USE_ERROR always results from attempts to CREATE files.

CE3107A - Test terminates when unhandled STATUS_ERROR results from attempt to CLOSE an unopened file.

CE3108A, CE3108B - Test terminates when unhandled USE_ERROR results from as attempt to CREATE a file.

CE3112B - Test incorrectly reports failure when USE_ERROR results from attempt to OPEN file.

CE3114A, CE3115A - Test incorrectly reports failure when USE ERROR results from attempt to CREATE files.

CE3114B - Test incorrectly reports failure STATUS_ERROR results from attempts to DELETE unopened files.

CE2102D, CE2102E - Test incorrectly reports failure when STATUS ERROR results from attempts to RESET unopened files.

CE2103A, CE2103B - Test terminates when unhandled STATUS_ERROR results from attempts to CLOSE an unopened file.

CE2104A, CE2104B - Test terminates when unhandled USE_ERROR results from attempt to CREATE a file.

CE3305A, CE3706F - Test incorrectly reports failure when its explicitly raised INCOMPLETES are handled as OTHERS.

EE3102C - Test terminates when unhandled USE_ERROR results from attempt to CREATE a file.

CZ1103A - Test incorrectly reports failure when USE_ERROR results from attempts to CREATE files.

4.2.8 Information Derived from the Tests

Processing of the following tests indicated support as described below for a variety of implementation options examined by the tests.

. E24101A-B.TST: If a based integer literal has a value exceeding SYSTEM.MAX__INT, an implementation may either reject the compilation unit at compile time or raise NUMERIC__ERROR at run time. This test showed that the compiler did not reject the compilation unit at compile time.

. B26005A.ADA: This test contains all the ASCII control characters in string literals. The system replaced the control characters corresponding to format effectors with a space in the listing file. All occurrences were identified with a diagnostic message by the compiler.

. D29002K-B.ADA: This test declares 713 identifiers and was passed by the compiler.

. E36202A-B.ADA and E36202B-B.ADA: These tests declare multidimensional null BOOLEAN arrays in which LENGTH of one dimension exceeds INTEGER'LAST and SYSTEM.MAX_INT, respectively. An implementation can accept this, or it can raise NUMERIC__ERROR or STORAGE_ERROR at run time. The compiler did accept the declarations and raised NUMERIC_ERROR during execution.

. D4A002A-AB.ADA and D4A002B.ADA: These tests contain universal integer calculations requiring 32 and 64 bits of accuracy, i.e., values that exceed SYSTEM.MAX_INT are used. An implementation is allowed to reject programs requiring such calculations. The compiler passed these tests.

. E43211B-B.ADA: If a bound in a non-null range of a non-null aggregate does not belong to an index subtype, then all choices may or may not be evaluated before CONSTRAINT__ERROR is raised. The compiler did not evaluate all choices before CONSTRAINT_ERROR is raised.

. E43212B-B.ADA: This test examines whether or not all choices are evaluated before subaggregates are checked for identical bounds. The compiler evaluates all subaggregates for identical bounds.

- E52103Y-B.ADA, C52104X-B.ADA, C52104Y-B.ADA: These tests declare BOOLEAN arrays with INTEGER'LAST+3 components. An implementation may raise NUMERIC_ERROR at the type declaration or STORAGE__ERROR when array objects of these types are declared, or it may accept the type and object declarations. The compiler did not raise NUMERIC_ERROR for null array with one dimension of length greater than INTEGER'LAST in E52103Y-B.

- A series of tests (D55A03*-AB.ADA) checks to see what level of loop nesting is allowed by an implementation. Tests containing up to 65 nested loops passed without exceeding the implementation's capacity.

- D56001B-AB.ADA contains blocks nested 65 levels deep. This test was passed.

- C94004A-B.ADA: This test checks to see what happens when a library unit initiates a task and a main program terminates without ensuring that the library unit's task is terminated. This test showed that such library tasks continued to execute even after the main program terminates and then terminated appropriately by themselves.

- CA1012A4M-B.DEP: This test checks whether an implementation requires generic library unit bodies to be compiled in the same compilation as the generic declaration. The compiler does allow generic declarations and bodies to be compiled in completely separate compilations.

- CE2106A-B.DEP and CE3110A-B.DEP: These tests confirm that dynamic creation and deletion of files is supported.

- CE3111A-B.DEP showed that two internal files may read the same external file.

- CE3111B-B.DEP and CE3111C-B.DEP showed that the compiler does allow two internal TEXT_IO files to be associated with the same external file when one or both internal files are opened for writing.

5. SUMMARY AND CONCLUSIONS

ACVC version 1.6 comprises 2162 tests, of which 63 were withdrawn due to errors. The Federal Software Management Support Center (FSMSC), an AVF, identified 1880 of the remaining tests to be applicable to SofTech, Inc.'s Ada86 compiler. The compiler passed all of these tests, compiled on the VAX-11/785, VAX-11/780, VMS 4.1 targeted to the INTEL 8086 and 80186.

The FSMSC considers these results to show acceptable conformity to the Ada Language Standard.

END

DTIC

6-86